

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-269998

(43)Date of publication of application : 29.09.2000

(51)Int.Cl.

H04L 12/40  
G06F 13/00  
H04L 12/28  
H04Q 9/00

(21)Application number : 11-075004

(71)Applicant : HITACHI LTD

(22)Date of filing : 19.03.1999

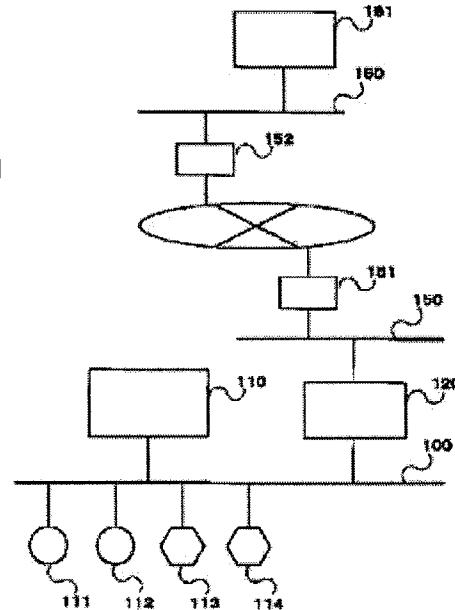
(72)Inventor : AIZONO TAKEO  
KONO KATSUMI

## (54) SETTING METHOD FOR DECENTRALIZED SYSTEM

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To facilitate setting and information management of devices by automatically recognizing devices connected via networks, gathering information that the devices holds, generating information for setting the devices according to the gathered information, and setting the generated information in the devices and actualizing a communication between the devices.

**SOLUTION:** A controller 110 communicates with devices via a network 100 for control, for example, conveyors in a production line. A computer 120 is connected even to a network 150. The network 150 is connected to the Internet via a gateway 151 and is able to communicate with a computer 161 connected to another network 160 through another gateway 152. The network 150 and network 160 are networks, such as Ethernet(R) computer, which allow computers to communicate with each other. Here, the controller 110 may be any computer which can execute a control program.



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2000-269998

(P2000-269998A)

(43)公開日 平成12年9月29日 (2000.9.29)

| (51)Int.Cl. <sup>7</sup> | 識別記号  | F I          | テマコード(参考)         |
|--------------------------|-------|--------------|-------------------|
| H 04 L 12/40             |       | H 04 L 11/00 | 3 2 0 5 B 0 8 9   |
| G 06 F 13/00             | 3 5 7 | G 06 F 13/00 | 3 5 7 A 5 K 0 3 2 |
| H 04 L 12/28             |       | H 04 Q 9/00  | 3 0 1 D 5 K 0 3 3 |
| H 04 Q 9/00              | 3 0 1 | H 04 L 11/00 | 3 1 0 D 5 K 0 4 8 |

審査請求 未請求 請求項の数 5 OL (全 14 頁)

|          |                        |         |  |
|----------|------------------------|---------|--|
| (21)出願番号 | 特願平11-75004            | (71)出願人 | 000005108<br>株式会社日立製作所<br>東京都千代田区神田駿河台四丁目 6 番地   |
| (22)出願日  | 平成11年3月19日 (1999.3.19) | (72)発明者 | 相薦 岳生<br>神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内 |
|          |                        | (72)発明者 | 河野 克己<br>神奈川県川崎市麻生区王禅寺1099番地 株式会社日立製作所システム開発研究所内 |
|          |                        | (74)代理人 | 100068504<br>弁理士 小川 勝男                           |
|          |                        |         |  |

最終頁に続く

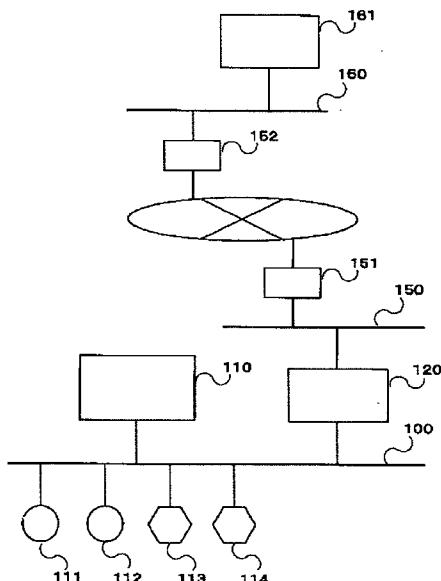
(54)【発明の名称】 分散システムの設定方法

(57)【要約】

【課題】本発明は、複数の機器をネットワークに接続して構成する分散システムにおいて、自動的に機器を設定して相互に通信を行うことを可能にする分散システムの設定方式に関する。

【解決手段】複数の機器をネットワークに接続して構成した分散システムにおいて、ネットワークに接続された機器を自動的に検出するステップと、検出された機器より機器が保持する情報を読み込むステップと、読み込んだ情報をもとにシステム構成を判断して通信を行うのに必要な構築情報を生成するステップと、生成された構築情報を各機器に設定するステップを実行することにより、機器間の通信設定を自動的に行うことを可能とする。

図1



## 【特許請求の範囲】

【請求項1】ネットワークに接続された複数の機器により構成される分散システムの設定方法であって、ネットワークに接続された機器を検出するステップと、検出された機器より情報を収集するステップと、収集された情報をもとに機器に設定する情報を判断するステップと、判断された情報を機器に設定するステップとからなる分散システムの設定方法。

【請求項2】請求項1記載の分散システムの設定方法において、ネットワークに計算機を接続し、該計算機にインストールされた構築ツールにより、機器を設定する分散システムの設定方法。

【請求項3】請求項2記載の分散システムの設定方法において、構築ツールが機器に関する情報を持たず、機器より収集した情報のみに基づいて機器の設定を行う分散システムの設定方法。

【請求項4】請求項1記載の分散システムの設定方法において、ネットワーク上にノードアドレスを格納したメッセージをブロードキャストし、該メッセージのノードアドレスと同じノードアドレスが設定されたノードが応答を送信し、該応答を受信することにより該ノードアドレスが設定されたノードがネットワークに接続されていると判断することにより、ネットワーク接続されたノードを自動的に検出する分散システムの設定方法。

【請求項5】ネットワークに接続された複数の機器により構成される分散システムの設定方法であって、ネットワークに接続された機器を自動的に検出するステップと、検出された機器より情報を収集するステップと、収集された情報をもとに機器に設定する情報を判断するステップと、判断した設定情報が間違っているときにはユーザインターフェースより入力された情報に基づいて修正するステップと、修正された情報を機器に設定するステップとからなる分散システムの設定方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、ネットワークに接続された複数の機器と計算機により構成される分散システムにおいて、これらのノード間でデータを授受するのに必要な各ノードの設定を行う分散システムにおける設定方法に関する。

## 【0002】

【従来の技術】ネットワークに複数の計算機と少なくとも1つのプリンタが接続され、計算機にインストールされたアプリケーションプログラムより印刷要求があると、該計算機に接続されたローカルプリンタが存在するか否かを判断したのち、ローカルプリンタが存在しない場合には該計算機が自内のファイルを読み込んでネットワークに接続されたリモートプリンタの接続処理を自動的に行うパソコンネットワークにおけるプリンタ接続のための設定方式が、例えば特開平6-242903にお

いて示されている。

## 【0003】

【発明が解決しようとする課題】従来の設定方式により機器の設定を行った場合、該機器と通信を確立するためには該機器に関する情報が必要であった。例えば、新しい機器をネットワークに接続した場合、該機器と通信するためには、機器の開発メーカーより通信ドライバなどのソフトウェアやデータを入手して、該機器と通信するための接続処理を行わなければならなかった。このため、多数の機器がネットワークに接続されるシステムでは入手しなければならない機器情報が増え、該情報の入手や管理が困難になるといった問題が生じた。特に、機器の追加やリプレースが頻繁に発生するシステムでは、頻繁に機器情報を入手し、これらの情報の追加や入れ替えも頻繁に行わなければならず、機器情報の管理が一層困難になる。また、管理している機器情報を更新せずに、機器の取り外しや変更を行うことにより、実際にネットワークに接続された機器とこれらの機器に関する機器情報とが一致しなくなるといった問題が生じ、システムの管理を非常に困難なものにしていた。

【0004】本発明では、多数の機器が接続された分散システムにおいて、機器の設定と機器の情報管理が困難であるという課題を解決することを目的とする。

## 【0005】

【課題を解決するための手段】上記課題を解決するため、本発明における設定方式では、(1)ネットワーク接続された機器を自動的に認識し、(2)認識された該機器から該機器が保持する情報を収集し、(3)収集された該情報に基づいて機器の設定を行うに必要な情報を生成し、(4)生成された該情報を機器に設定することにより機器間の通信を実現すること、を特徴とする。

## 【0006】

【発明の実施の形態】本発明における第1の実施例を説明する。

【0007】本発明における第1の実施例では、生産システムにおける機器の設定方式を例に説明する。図1は本発明におけるシステムの構成例である。コントローラ110と機器111、機器112、機器113、機器114がネットワーク100に接続されている。機器111と機器112はセンサーなどの生産ラインの情報を収集する入力機器であり、機器113と機器114はモーター制御などを行う生産ラインを動かす出力機器である。コントローラ110は、ネットワーク100を介してこれらの機器と通信を行い、例えば生産ラインのコンベアを制御する。このシステムでは、コントローラ110により機器を集中制御する。計算機120は、ネットワーク100に接続されたノードを設定するための構築ツールをインストールしており、計算機120の構築ツールはネットワーク100を介して、コントローラ110、機器111、機器112、機器113、機器114

を設定する。計算機120は、ネットワーク150にも接続されている。ネットワーク150は、ゲートウェイ151を介してインターネットに接続され、他のゲートウェイ152を介して他のネットワーク160に接続された計算機161と通信することができる。ネットワーク150とネットワーク160は、例えばイーサネットなどの計算機間で通信を行うためのネットワークである。ここで、コントローラ110は制御プログラムを実行できるものであればよく、計算機であってもよい。また、機器と計算機を接続するネットワーク100はノード間で相互にデータを交換できる手段であればバス型のネットワークでなくてもよく、無線通信などの手段を用いてもよい。

【0008】ネットワーク100に接続された機器とコントローラの構成例を図2示す。ネットワーク200に接続された機器250は、通信や機器の制御を行う制御部260と、機械的あるいは電気的に機器を駆動する機械部270により構成される。例えば、光センサーの場合、機械部270は赤外線などを送信して物体を検出する部分であり、検出結果を電気信号として制御部260に伝達する。制御部260は、この電気信号をデジタル信号に変換したり、工学値に変換するなどの処理を行い、メッセージにこの処理結果を格納してネットワーク200に送信する。制御部260は、ネットワーク100上の信号を処理するトランシーバ206、メッセージの組み立てなどの通信制御を行う通信プロセッサ201、機器の制御プログラムを実行するプロセッサ202、通信や機器の制御プログラムと種々のデータを格納するための不揮発性記憶装置であるROM203、プログラムを実行する際に使用する揮発性記憶装置であるRAM204、機械部270とデータを交換するためのI/Oインターフェース205により構成される。I/Oインターフェース205には、I/O(207)が接続され、制御部260はこのI/Oインターフェース205を介して機械部270からの信号の入力および機械部270への信号の出力を行う。プロセッサ201、プロセッサ202、ROM203、RAM204、I/Oインターフェース205は内部バス210に接続されており、相互にデータを交換することができる。コントローラは、プロセッサ202の処理能力、ROM203とRAM204の記憶容量が機器と異なる。またI/Oインターフェース205と機械部270は備えていない。他の構成は機器250と同じである。

【0009】ネットワーク100を介して各ノードが送受信するメッセージのフォーマットを図3に示す。メッセージ300は、識別子301、アドレス302、データ303により構成される。識別子301はメッセージを識別するために付与される文字列であり、アドレス302はネットワーク100に接続された各ノードのノードアドレスであり、データ303はセンサーの測定結果

やモーターを制御するための制御データなどである。アドレス302には、メッセージを送信したノードのノードアドレス、あるいはメッセージを送信する先のノードアドレスなどが設定される。メッセージ300を受信したノードは、メッセージの識別子301により、アドレス302に設定されるノードアドレスが何を示すかを判断する。ネットワーク100に接続された各ノードは、メッセージ300に付与された識別子301、あるいは識別子301とアドレス302により、メッセージ300を受信するか否かを判断する。

【0010】各ノードの設定を行う構築ツールは、ネットワーク100に接続されたノードを自動的に認識する。このときのデータの流れを図4に示す。構築ツールは処理を開始すると、まずネットワークに接続されているノードを全て検出する。図4(a)に示すように、ネットワーク400にコントローラ410、入力機器411、入力機器412、出力機器413、構築ツール430をインストールした計算機420が接続されている。ネットワーク400に接続される各ノードには重複しないようノードアドレスが付与される。ノードアドレスは、デイプスイッチまたはローカルな端末などにより各ノードに事前に設定される。構築ツール430は、起動されるとメッセージ401をネットワーク400に送信する。メッセージ401には、識別子301としてノード確認メッセージであることを示す識別子Aが付与され、アドレス302には構築ツールが適当なノードアドレスを設定する。データ303は付与されない。

【0011】ネットワーク400に接続された各ノードはメッセージ401を受信すると、メッセージ401の識別子301によりメッセージ401がノード確認メッセージであると判断する。これにより、ノード確認メッセージに適した処理を実行する。各ノードはメッセージ401内のアドレス302を読み出し、自ノードのノードアドレスと比較する。もしも、メッセージ401内のアドレス302と自ノードのノードアドレスが一致した場合、アドレス302に自ノードアドレスを設定し、メッセージ確認メッセージであることを示す識別子Aを付与して応答メッセージ402をネットワーク400に送信する。構築ツール430は、他ノードからの応答メッセージ402を受信することにより、自分がメッセージ401に設定したノードアドレスと同じノードアドレスを持つノードがネットワーク400に接続されていると判断する。図4(a)の例では、構築ツール430がノード確認メッセージ401にノードアドレス1番を設定してネットワーク400にブロードキャストしている。これを受信した各ノードは自ノードのノードアドレスがノードアドレス1番であるか否かを判断する。入力機器411には、ノードアドレス1番が割り当てられており、入力機器411は自ノードのノードアドレス1番を応答メッセージ402に設定してブロードキャストす

る。構築ツールは、ノードアドレス1番が設定された応答メッセージ402を受信することにより、ノードアドレス1番が設定されたノードがネットワーク400に接続されていることを検出する。構築ツール430は、ノードアドレスの最小値からノードアドレスの最大値までの全てのアドレスに対して同じ処理を順次繰り返すことにより、ネットワークに接続されている全てのノードのノードアドレスを検出することができる。

【0012】また、定期的にこの処理を繰り返してネットワークに接続されている全ノードを検出し、新たに検出したノードと前回検出したノードとを比較することにより、ネットワークから切り離されたノードを検出できる。ここで、図3にて示したメッセージの識別子やアドレスのサイズは事前に決まっており、例えば各2バイトである。ノード確認メッセージであることを示す識別子Aも事前に決まっており、例えば数次の1000で表される。各ノードはメッセージの最初の2バイトに1000が付与されたメッセージを受信すると、そのメッセージがノード確認メッセージであると判断する。ノード確認メッセージのメッセージフォーマットは事前に定義されており、ノード確認メッセージではアドレスとして2バイトのノードアドレスが設定され、データは存在しないことが事前に決まっている。新しいノードがネットワーク400に接続されると、構築ツール430はこのノードを自動的に検出する。図4(b)に示すように、図4(a)のシステム構成に新たに出力機器414が追加された場合、出力機器414は自ノードのノードアドレスにノード確認メッセージであることを示す識別子Aを付与したメッセージ403をネットワーク400にブロードキャストする。構築ツール430は、このメッセージを受信することにより、新しくノードが追加されたことを検出し、そのノードのノードアドレスを知ることができる。新たにネットワーク400に接続されたノードは、必ず自ノードのノードアドレスを設定したノード確認メッセージを送信する。

【0013】構築ツール430は、ネットワーク400に接続されたノードのノードアドレスを検出すると、そのノードアドレスを使い、そのノードにメッセージを送信する。メッセージのアドレスとして送信先ノードアドレスを指定してメッセージを送信することにより、そのノードに対してメッセージを送信することができる。そのノードに対してデータ送信を要求することにより、ノードが保持する情報を読み込む。機器を生産するメーカーは、機器の出荷時に機器に関する種々の情報をROM203に格納している。機器に格納されたこれらの情報は、ネットワーク200に接続された他のノードからトランシーバ206を介してプロセッサ201にデータ読み込み要求を送信することにより読み込むことができる。これらの情報へのアクセス方式は、標準規格などにより予め決められており、各メーカーから提供される機

器の情報を同じように読み込むことが可能である。例えば、機器を開発したメーカー名は30バイトで表現し、その情報の番号が3番であることが予め決まっている。機器Aの開発メーカー名をネットワーク経由で読み込むためには、識別子としてデータを要求するメッセージであることを示す識別子、アドレスとして機器Aのノードアドレス、そしてデータとして情報の番号を示す3というデータ(情報番号3番を指定)を設定してメッセージをブロードキャストする。これを受信した機器Aの通信プロセッサ201は、ROM203より開発メーカー名を読み出す。データを要求するメッセージの応答メッセージであることを示す識別子、自ノードのノードアドレス、データとして読み出した開発メーカー名をメッセージに設定してネットワークにブロードキャストする。

【0014】構築ツールは識別子とアドレスをもとに、この応答メッセージを受信し、機器Aの開発メーカー名を受信することができる。このアクセス方式を用いて構築ツール430は検出したネットワークに接続される全てのノードからノードが保持する情報を読み出す。読み出された情報は、図5に示したノード管理テーブル500に登録される。ノードアドレス501には検出したノードのノードアドレスが登録され、これらのノードより読み出した情報を順次ノード管理テーブル500に登録する。入出力区分502は、ノードが入力機器であるか、出力機器であるかを示す情報であり、データ長503はノードが送信または受信できるデータのデータ長であり、メーカー504はノードを開発したメーカーのメーカー名であり、ファイルアドレス505は機器の情報が十分である場合にアクセスするファイルのインターネット上のアドレスであり、カタログ番号506はノードのカタログ番号であり、時間間隔507はノードがメッセージを送信する時間間隔またはメッセージを受信処理可能な時間間隔である。センサーのような機器の場合、コストや実装スペースの問題から計算機のように十分なROMを備えていない。このため、構築ツールが必要とする全てのデータをROMに格納しているとは限らない。したがって、構築ツールが必要とする情報を機器が保持していない場合もある。

【0015】生産ラインのようなシステムを動かすには、センサーなどの入力機器からの情報を読み込んで処理し、その処理結果をモーターなどの出力機器に送つてモーターを駆動する。これを実行するには、センサーのプログラム、コントローラの制御プログラム、モーターのプログラム間で通信を行う必要がある。つまり、プログラム間の通信設定を行わなければならない。構築ツールはノード管理テーブルに登録された情報をもとに、通信設定を行うのに必要な構築情報を生成する。構築ツールの設定例を図6に示す。図6に示したシステムは、コントローラ610、入力機器620、出力機器630により構成され、これらのノードはネットワーク600

7  
 を介してメッセージを送受信する。このネットワーク600には、構築ツールをインストールした計算機も接続されているが図中には記していない。コントローラ610にはプログラムがインストールされている。これらのプログラムは、機器を制御するための制御プログラムであるアプリケーションプログラム611と、通信を制御するプログラムであるプロトコル処理プログラム612である。同様に入力機器620には、入力機器を制御するアプリケーションプログラム621とプロトコル処理プログラム622がインストールされており、出力機器630には、出力機器を制御するアプリケーションプログラム631とプロトコル処理プログラム632がインストールされている。これらのプログラムはROM203からプロセッサ202によりRAM204にインストールされるか、あるいはネットワークよりRAM204に直接インストールされる。入力機器620のアプリケーションプログラム621にて測定されたデータは、プロトコル処理プログラム622を介してネットワーク600に送信される。このデータはコントローラ610のプロトコル処理プログラム612にて受信され、アプリケーションプログラム611にて処理される。処理結果はプロトコル処理プログラム612によりネットワーク600に送信され、出力機器630のプロトコル処理プログラム632にて受信される。受信された処理結果は、アプリケーションプログラム631に渡され、出力機器630はこの処理結果をもとにモーターの回転数を変更するなどの制御を行う。

【0016】構築ツールは、構築情報に基づいてプロトコル処理プログラム612、622、632に情報を設定し、このデータの流れを実現する。情報の設定方法は、各ノードの情報へのアクセス方式と同様に、標準規格などにより予め決まっているものとする。入力機器620のプロトコル処理プログラム622には、構築ツールにより送信するメッセージの識別子、メッセージの送信時間間隔などが設定される。コントローラ610のプロトコル処理プログラム612には、入力機器620が送信したメッセージを受信するために必要な情報として、入力機器620が送信するメッセージの識別子、入力機器620のノードアドレス、メッセージのデータ長、メッセージの送信時間間隔などが設定される。これらの情報を設定することにより、入力機器620が送信したメッセージをコントローラ610が識別して受信することができ、通信が確立する。送信時間間隔は、入力機器620がメッセージを送信する時間間隔であり、コントローラ610では入力機器620のタイムアウト監視などを行うのに使用する。プロトコル処理プログラム612により受信されたメッセージは、アプリケーションプログラム611にて受信される。

【0017】アプリケーションプログラムは、`receive(id, address, data)`といった

10  
 関数を使って受信メッセージを受け取る。`id`はメッセージの識別子であり、例えばセンサーの情報であることを示す。`address`は入力機器620のノードアドレスであり、例えどもセンサーが測定した情報であるかを示す。`data`は、例えばセンサーの測定結果である。アプリケーションプログラム611は、入力機器620からのデータを処理し、出力機器630に送信する。コントローラ610から出力機器630へメッセージを送信するには、入力機器620からコントローラ610にメッセージを送信したときと同様に、コントローラ610のプロトコル処理プログラム612と出力機器630のプロトコル処理プログラム632に構築ツールが必要な情報を設定する。コントローラ610から出力機器630にメッセージを送信する場合、コントローラ610は出力機器630の時間間隔507以上の時間間隔を空けてメッセージを送信することにより、出力機器630の受信頻度が高まり、メッセージを受信しきれなくなるという状況を回避する。構築ツールがプロトコル処理プログラムに設定するこれらの構築情報は、ノード管理テーブル500より構築ツールが自動的に生成する。

20  
 30  
 40  
 50  
 【0018】以下、この生成方式を説明する。まずノードの入出力区分502により、ノードがコントローラであるか、入力機器であるか、出力機器であるかを判断する。コントローラである場合には、入力と出力が設定されている。次にデータ長503を読み込み、入力機器が送信するデータ長と出力機器が受信するデータ長を判断する。コントローラの送受信データ長は、機器側のデータ長に合わせて設定する。例えば、入力機器の送信データ長が2バイトであった場合、コントローラの受信データ長は2バイトにする。さらに、時間間隔507を読みこむ。時間間隔507も機器に合わせてコントローラを設定する。例えば、入力機器の時間間隔507が100ミリ秒であった場合、コントローラには受信時間間隔100ミリ秒が設定され、この時間間隔をもとに入力機器のタイムアウト監視を行う。出力機器の時間間隔507が500ミリ秒であった場合、出力機器の受信限界性能が500ミリ秒であるため、コントローラには送信時間間隔として500ミリ秒が設定され、コントローラは500ミリ秒以上の時間間隔を空けて出力機器にメッセージを送信する。コントローラの設定を機器に合わせるのは、コントローラのプロセッサ性能の方が高く、ROMやRAMといったリソースを十分持っているからである。さらにコントローラと機器間の通信に識別子を割り当てる。メッセージの識別子が、ノード確認メッセージを示す識別子やノード情報の読み込みを要求するメッセージであることを示す識別子のように、事前に決められていない場合、構築ツールが自動的に識別子を割り当てる。コントローラが入力機器Aと入力機器Bからメッセージを受信して、出力機器Cと出力機器Dにメッセージ

を送信する場合、例えばメッセージの識別子を番号で表し、1番から順に番号で割り当てる。つまり、入力機器Aからコントローラに送信するメッセージには識別子1を割り当て、入力機器Bからコントローラに送信するメッセージには識別子2を割り当てる。以上のようにして、構築情報が生成される。例えば、入力機器Aでは送信するメッセージの識別子は1、2バイトのデータを100ミリ秒間隔で送信するといった構築情報が生成される。入力機器Aとコントローラのプロトコル処理プログラムに設定する。またコントローラ側に関しては、識別子1とノードアドレス1番が付与されたメッセージを受信し、データのデータ長は2バイトであり、100ミリ秒周期で受信するという構築情報を生成する。これらの構築情報を入力機器Aとコントローラに設定することにより、入力機器Aより送信されたメッセージをコントローラにて受信できる。

【0019】構築ツールの処理フローを図7と図8に示す。構築ツールの処理は大きく4つのステップに分けられる。

【0020】(1) ネットワークに接続されたノードを自動的に検出するステップ1。

【0021】(2) 検出したノードより情報を読み込むステップ2。

【0022】(3) 読み込んだ情報をもとに構築情報を生成するステップ3。

【0023】(4) 生成された構築情報を各ノードに設定するステップ4。

【0024】図7にステップ1とステップ2の処理を、図8にステップ3とステップ4の処理を示す。

【0025】構築ツールは起動すると、ノードのノード番号を表す変数Nに0を設定して初期化し(ステップ701)、ノードアドレスとしてNを設定したノード確認メッセージをネットワークに送信する(ステップ702)。もしも、ノード確認メッセージの応答メッセージを受信できた場合、つまりノードアドレスNのノードがネットワークに接続されていた場合(ステップ703)、ノードアドレスNをノード管理テーブルに登録する(ステップ704)。もしも一定時間応答が得られなかった場合(ステップ703)、ノードアドレスNはノード管理テーブルに登録されない。ノードアドレスNが最後のノードアドレスでなければ(ステップ705)、ノードアドレスNに1を加え(ステップ706)、次のノードアドレスに対して同じ処理を繰り返す。もしも最後のノードアドレスであった場合(ステップ705)、ノード管理テーブルに登録されたノードアドレスを1つ読み出し(ステップ707)、読み出したノードアドレスが示すノードよりネットワークを介して必要な情報を読み込む(ステップ708)。ノード管理テーブルに登録された全てのノードアドレスが示すノードに対して情報を読み込む処理を終了しているか否かを判断する(ス

テップ709)。全てのノードから読み込んでいなかった場合、次のノードアドレスを読み込んで同じ処理を繰り返す。ノード管理テーブルに登録された全てのノードアドレスが示すノードより情報を読み込む処理を終了した場合(ステップ709)、図8に示した次のステップに移行する。

【0026】ネットワークに接続したノードより収集した情報を登録したノード管理テーブルをもとに、まずマスターノードを決める(ステップ801)。マスターノードとは、機器を制御するコントローラや計算機であり、入出力区分502が入出力であることから判断される。スレーブとは、マスターノードにより制御される機器である。本実施例におけるシステムは集中制御システムであり、マスターノードとなりうるノードは1つしか存在しない。もしも、マスターノードとなりうるノードが2つ以上存在した場合(ステップ802)、つまり入出力区分502に入出力が登録されたノードが2つ以上存在した場合、オペレータに対して異常を通知し(ステップ807)、処理を終了する。マスターノードが識別できた場合(ステップ802)、スレーブノードの通信条件を確認する(ステップ803)。つまり、入出力機器のデータ長や時間間隔など、構築情報を生成するのに必要な情報が全て揃っているか否かを判断する。もしも必要な情報が揃っていない場合(ステップ804)、インターネットにアクセスして不足している情報を収集することを試みる(ステップ805)。各機器は保持できる情報量が限られているため、不足している情報を補うデータを格納したファイルにアクセスするためのファイルアドレス505を保持している。このファイルは、例えば機器を開発したメーカーが自社のサーバー上に保持している。ファイル名は、例えばカタログ番号506である。ファイルアドレス505は、例えばftp://www.hitachi.co.jp/Device/Data/であり、構築ツールはこのファイルアドレス505をもとにインターネットを介して図1で示された開発したメーカーのファイルサーバーである計算機161にアクセスする。機器に関する情報はカタログ番号506と同じ名称のH-X Y3.TXTというテキストファイルで保存されている。このファイルフォーマットは各社同じであり、標準規格などで予め規定されている。構築ツールは、例えば機器が時間間隔507を保持していなかった場合、ファイルアドレス505で示されたアドレスよりファイルを読み込んで時間間隔507をノード管理テーブル500に登録する。

【0027】もしも、ファイルが存在しないなどの理由により、構築情報を生成するのに必要なスレーブノードの情報が入手できなかった場合(ステップ806)、ノードの構築が行えないことをオペレータに通知し(ステップ807)、処理を終了する。もしも、必要な情報が入手できた場合には(ステップ806)、処理を継続す

る。構築情報を生成するのに必要なスレーブ情報が揃った場合（ステップ804、ステップ806）、図6で説明したように、構築情報を生成する（ステップ808）。生成された構築情報をもとにマスターノードのリソースが十分であるかを判断する。もしもリソースが不足している場合（ステップ809）、オペレータに対しリソースが不足していることを通知し（ステップ807）、処理を終了する。例えば、通信を行う機器の数が多く、マスターノードのプロトコル処理プログラムを実行するのに必要なメモリ容量が足りない場合、リソースが不足していると見なす。最後に、生成した構築情報を各ノードに設定し（ステップ810）、処理を終了する。なお、ここでは必要な情報が不足している場合、インターネットを介してファイルを読み込む例を示したが、この情報は構築ツールがインストールされている計算機のローカルなデータベース上にあってもよい。この場合、ファイルアドレス505は、例えば `f i l e : / / c : ¥ / D e v i c e / D a t a /` となりローカルなディレクトリを指定する。

【0028】図7と図8では、ネットワークに接続しているノードを自動的に検出し、これらのノードからノードが保持する情報を読み出し、これらのノードより読み込んだ情報をもとに構築情報を生成し、ノードを設定する方式を説明した。しかし、実際のシステムでは一旦ノードを設定したあとでシステムの拡張に伴うノードの追加が発生したり、機器の劣化に伴うリプレースが発生し、システムの状態は常に変化する。システム構成が変化により追加・変更されたノードを自動的に設定する方式がもとめられる。追加されたノードを自動的に検出して構築する処理を表す処理フローを図9に示す。図9の処理は、図7と図8で示した処理が終了したのち、継続して行われる。図4（b）で説明したように構築ツールは常にネットワークを流れるメッセージを受信し、ノード確認メッセージが流れていなければそれを判断している。構築ツールは、ノード確認メッセージを受信すると（ステップ901）、ノード管理テーブル500より受信したノード確認メッセージに設定してあるアドレスと同じノードアドレスがすでにノード管理テーブル500に登録されているか否かを判断する。もしも、すでに同じアドレスが存在している場合（ステップ902）、追加されたノードのノードアドレスの設定が間違っているため、ノードの構築が行えない。ノードアドレスの設定が間違っていることをオペレータに通知し（ステップ903）、処理を終了する。アドレスが重複していない場合（ステップ902）、ノード管理テーブルにノードアドレスを追加する（ステップ904）。そして、ノードアドレスが示すノードより情報を収集してノード管理テーブルに登録する（ステップ905）。必要な情報が収集できたかどうかを判断し（ステップ906）、もしも必要な情報がない場合には（ステップ907）、インター

ネットにアクセスして（ステップ908）、不足している情報をファイルより読み込むことを試みる。もしも必要な情報を入手できない場合には（ステップ909）、必要な情報が入手できることをオペレータに通知し（ステップ903）、処理を終了する。もしも必要な情報が揃った場合（ステップ907、ステップ909）、構築情報を生成する（ステップ910）。新しくノードが追加されたことにより、マスターノードのリソースが不足するか否かを判断し、もしもリソースが不足する場合には（ステップ911）、その旨をオペレータに通知して（ステップ903）、処理を終了する。もしも、リソースが不足しない場合には（ステップ911）、各ノードに構築情報を設定し（ステップ912）、処理を終了する。

【0029】機器のリプレースは、ノードのネットワークからの切り離しとノードの追加という2つの処理が発生したと捉える。ノードの追加が行われた場合、図9に示したノード追加時の設定方式を用いればよい。切り離されたノードの検出は、先に説明した通り図7に示したネットワークに接続されたノードの自動検出を一定時間間隔で繰り返せばよい。ただし、自動検出されたノードのノードアドレスはノード管理テーブル500とは別のテーブルを準備して管理する。ノード管理テーブル500のノードアドレス501と新たに準備したテーブルに登録されたノードアドレスを比較し、前回の検出時に検出したノードと、新たに検出したノードを比較することにより、検出できなくなったノードが存在しないかを確認する。もしも、検出できなくなったノードが存在した場合、このノードはネットワークから切り離されたと見なし、ノード管理テーブル500に登録されているこのノードに関する情報を消去する。このようにして切り離されたノードの情報をノード管理テーブル500より消去したのち、図9に示した追加ノードの自動設定方式を用いれば、リプレースされた機器も自動的に設定することができる。

【0030】この実施例によれば、複数の機器を制御する1つのコントローラあるいは計算機と生産ラインを制御する複数の機器により構成される分散システムにおいて、構築ツールがネットワークに接続されたノードを自動検出するステップと、検出されたノードの情報を収集するステップと、この情報によりシステムの構築情報を生成するステップと、生成された構築情報を各ノードに設定するステップにより、分散システムを自動的に設定することができる。これにより、従来事前に通信ドライバや機器に関するデータファイルを準備しなければならなかった機器の通信設定が、事前情報を準備しなくても自動的に実行され、システム開発者の情報管理を行う負担が軽減される。また、通信ノウハウのないシステム開発者であってもシステムの構築を容易に行なうことができる、システム開発者の通信に関する学習期間と学習コスト

トが削減され、またシステム構築のための作業期間が大幅に短縮され、生産システム導入後すぐに生産を開始することができる。

【0031】本発明における第2の実施例を説明する。

【0032】本発明における第2の実施例は、集中制御を行うコントローラや計算機が存在しない点において第1の実施例と異なる。ネットワークに接続されたノードを自動検出する第1のステップと、検出されたノードより情報を収集する第2のステップは、第1の実施例と同じである。

【0033】第2の実施例では、家庭内の電化製品をネットワークで接続して制御するホームオートメーションシステムを例に説明する。システム構成を図10に示す。監視パネル1010、AV機器コントローラ1011、テレビ1012、ビデオ1013、オーディオ1014、照明スイッチ1(1015)、照明スイッチ2(1016)、照明1(1017)、照明2(1018)、照明3(1019)がネットワーク1000に接続され、ネットワーク1000を介して相互にデータを交換することができる。各機器は図2にて説明した機器250と同様に、通信インターフェースを備え、機器を制御するプログラムを内蔵する。AV機器コントローラ1011は、テレビやオーディオなどのAV機器を制御するコントローラであり、オーディオ1014はミニコンポのようなオーディオ機器である。監視パネル1010は、機器の動作状況を表示したり、機器を操作するためのコントローラであり、液晶ディスプレイのような表示装置を備える。監視パネル1010には構築ツール1050がインストールされており、この構築ツール1050が各ノードを構築する。ここでは、構築ツール1050が監視パネル1010にインストールされている例を示したが、表示装置を備え、ネットワークに接続されているノードであれば、テレビやパーソナルコンピュータなどの他の機器や計算機にインストールされていてもよい。また、ここではネットワーク1000としてバス型のネットワークを使った例を示したが、データを送受信できる手段であれば無線通信や電力線を使った通信を用いてもよい。構築ツール1050がネットワーク1000に接続されたノードを自動的に検出するステップと、検出されたノードより情報を読み込むステップは第1の実施例と同じである。しかし、ノードより読み出す情報は異なる。

【0034】第2の実施例におけるノード管理テーブルを図11に示す。ノードアドレス1101は、ネットワークより自動的に検出されたノードのノードアドレスである。機器種別1102は、AV機器、照明機器、空調機器、セキュリティ機器といった機器の種類を表す。入出力区分1103、データ長1104、メーカー1106は、第1の実施例と同じである。設定位置1105は、機器の設置された場所を示す。機器名称1107

は、機器の名称を表し、例えば「インバーターエアコン白くまくん」といった製品名称である。ノードアドレス1101、機器種別1102、入出力区分1103、データ長1104、メーカー1106、機器名称1107は、製品出荷時にROMに格納される情報であり、第1の実施例と同じ方式により構築ツールは機器より読み込むことができる。ノードアドレス1101は、例えば8バイトのアドレスであり、最初の4バイトがメーカー番号、次の4バイトが各メーカーにて管理する製品番号である。メーカー番号を各メーカー間で重複しないように調整することにより、全ての機器に重複しないノードアドレスを出荷時に割り当てることができる。

【0035】設置位置1105は機器のユーザが機器を購入し、設置したあとでないと決めることができない。設置位置1105の取得方式を図12に示す。機器の設置位置は、各機器毎に自動的に取得し、自ノード内に保存する。家の中の部屋であるB室1200には、赤外線通信の発信機能を持つ位置発信機器1220、照明スイッチ1(1221)、照明1(1222)、照明2(1223)、照明3(1224)が設置されている。照明スイッチ1(1221)、照明1(1222)、照明2(1223)、照明3(1224)は赤外線通信の受信機を備え、赤外線通信により得られたデータを自内に読み込み、内部のRAMに格納する。位置発信機器1220は、位置発信機器1220出荷時に異なる識別子が割り当てられており、電源が供給されるとこの識別子を周期的に送信する。各部屋には、この位置発信機器1220が設置されており、異なる識別子を発信している。以下、この識別子を位置識別子と呼ぶ。照明スイッチ1(1221)、照明1(1222)、照明2(1223)、照明3(1224)は、位置発信機器1220が送信する位置識別子1250を受信し、ノード管理テーブル1100に設置位置1105として登録する。これにより、同じ部屋に設置された機器は同じ設置位置1105を保持することができる。

【0036】構築ツールの処理フローを図13に示す。ネットワークに接続されたノードを自動的に検出するステップと、検出されたノードより情報を読み出すステップは図7と同じである。構築ツールは、ノード管理テーブル1100の情報をもとに各機器の設定を行う。まず、各ノードを機器種別1102により分類する(ステップ1301)。次に、入出力区分1103を読み込み、機器種別により分類した機器グループ毎に構築情報の生成を試みる(ステップ1302)。各グループに入力機器が1つで出力機器が複数、あるいは入力機器が複数で出力機器が1つであれば、通信する相手は一意に決まる。このため、データ長1104などの情報をもとに第1の実施例と同じように各機器グループ毎に構築情報を生成できる。しかし、入力機器と出力機器が各々複数存在する場合には、通信の関係が一意に決まらず、機器

の設定を行えない。例えば、図10に示したシステム構成例では、AV機器はAV機器コントローラ1011、テレビ1012、ビデオ1013、オーディオ1014である。メッセージを送信する入力機器はAV機器コントローラ1011のみであり、他は全てメッセージを受信する出力機器である。

【0037】このため、AV機器コントローラ1011より、テレビ1012、ビデオ1013、オーディオ1014にメッセージを送信するという通信の関係が一意に決まり、各機器の設定を行うことができる。しかし照明機器は、照明スイッチ1(1015)、照明スイッチ2(1016)、照明1(1017)、照明2(1018)、照明3(1019)があり、入力機器が照明スイッチ1(1015)、照明スイッチ2(1016)の2つであるのに対し、出力機器は照明1(1017)、照明2(1018)、照明3(1019)の3つである。このため、どの照明スイッチが送信したメッセージをどの照明が受信するのかが判断できず、通信の関係を一意に定義することができない。このように、機器種別1102と入出力区分1103により通信の関係が一意に決まり構築情報を生成できる場合(ステップ1303)、構築情報を各ノードに設定し(ステップ1311)、処理を終了する。もしも、構築情報を生成できない場合

(ステップ1303)、さらに設置位置1105を使って通信の関係を判断する(ステップ1304)。機器種別1102、入出力区分1103、設置位置1105により、構築情報を生成できる場合(ステップ1306)、構築情報を各ノードに設定し(ステップ1311)、処理を終了する。例えば、図10に示した照明スイッチ1(1015)、照明スイッチ2(1016)、照明1(1017)、照明2(1018)、照明3(1019)の中で、照明スイッチ1(1015)と照明1(1017)が同じ部屋に設置されていて、同じ位置識別子が設置位置1105としてノード管理テーブル1100に登録されており、照明スイッチ2(1016)、照明2(1018)、照明3(1019)が同じ部屋に設置されていて、同じ位置識別子が設置位置1105としてノード管理テーブル1100に登録されている場合、同じ部屋に設置している機器間で通信を行うと判断して構築情報を生成する。これらの2つの部屋には入力機器が1つずつしかなく、一意に通信の関係を定義できる。しかし、これらの機器が同じ部屋に設置されている場合、どの照明スイッチとどの照明で通信を行うかが判断できない。このように構築情報を生成できない場合

(ステップ1306)、構築ツールはシステム構成を仮定して構築情報を生成する(ステップ1307)。

【0038】例えば、照明スイッチ1(1015)と照明スイッチ2(1016)が同数の照明を制御すると仮定し、照明1(1017)、照明2(1018)、照明3(1019)のうち、2つを照明スイッチ1(101

5)に、1つを照明スイッチ2(1016)にて制御すると判断する。ノードアドレスの小さい順に自動的に入力機器に対して出力機器を割り当てる。例えば、照明スイッチ1(1015)より照明1(1017)と照明2(1018)にメッセージを送信し、照明スイッチ2(1016)より照明3(1019)にメッセージを送信すると判断し、構築情報を生成する。しかし、このような仮定により判断した通信の関係が必ずしもユーザーの要求と一致するとは限らない。このため、ユーザーに通信の関係を提示し、判断結果が正しいか否かを確認する(ステップ1308)。ユーザーが正しいと判断した場合(ステップ1309)、生成した構築情報を各ノードに設置する(ステップ1311)。もしも、ユーザーが正しくないと判断した場合(ステップ1309)、ユーザーは通信の関係を変更する(ステップ1310)。構築ツールは、この変更結果により構築情報を修正し、この修正された構築情報を各ノードに設定し(ステップ1311)、処理を終了する。

【0039】ステップ1309におけるユーザーへの構築情報の確認画面の構成例を図14に、ステップ1310における構築情報の変更画面の構成例を図15に示す。図14は、ユーザに対して構築ツールが生成した構築情報を表示する構築情報表示画面1400である。ユーザーが構築情報が正しいか否かを判断するのに不要な識別子やメッセージのデータ長などは表示していないが、これらの情報を表示してもよい。構築情報表示画面1400では、ノードを識別するための情報としてノード管理テーブル1100に登録されているメーカー1106と機器名称1107を各機器毎に表示している。構築情報表示画面1400は、A社製AVリモコン1401がA社製Xテレビ1402、B社製Yビデオ1403、C社製Zカセットプレーヤー1404にメッセージを送信して制御することを示している。同様に、N社製Y照明スイッチ1405はH社製Y照明1407とJ社製Z照明1408にメッセージを送信して制御し、M社製X照明スイッチ1406はK社製X照明スイッチ1409にメッセージを送信して制御する。ここで、制御とは電源のオン／オフ制御などである。この内容が正しければ、ユーザーはマウスやタッチパネルなどの入力手段を用いて「はい」を選択し、この内容が正しくなければ「いいえ」を選択する。「いいえ」が選択されると図15に示す構築情報変更画面1500に移行する。

【0040】構築情報変更画面1500は、構築情報表示画面1400と同じ情報を表示しているが、構築内容を変更できる点が異なる。例えば、J社製Z照明をN社製Y照明スイッチが押されたときに点灯されるのではなく、M社製X照明スイッチが押されたときに点灯するようにならざる場合には、マウスやタッチパネルなどの入力手段により、M社製X照明スイッチを示す図(1551)とJ社製Z照明を示す図(1552)を選択する。

これにより、J社製Z照明が通信する相手がN社製Y照明スイッチからM社製X照明スイッチに変更される。構築ツールは、ユーザーにより終了ボタンが選択され、構築内容の変更作業が終了すると、ユーザーが変更した内容にしたがって構築情報を修正する。

【0041】この実施例によれば、複数の機器により構成される分散システムにおいて、構築ツールがネットワークに接続されたノードを自動検出するステップと、検出されたノードの情報を収集するステップと、この情報によりシステムの構築情報を生成するステップと、構築情報が生成できない場合にはユーザーが構築情報を変更するステップと、生成された構築情報を各ノードに設定するステップにより、分散システムを自動的に設定することができる。従来では、機器の通信設定を行うには通信に関する知識や機器自体に関する知識を必要としたが、これらの設定が自動的に行われるようになり、システムエンジニアなどのサポートなしにシステムを構築することができ、一般家庭などにもシステムを容易に導入できる。仮に、システムが複雑な構成をしているために自動設定が行えず、ユーザーによる設定補助を必要とした場合でも、ユーザーに分かりやすい形で構築内容を表示し、特別な通信や機器に関する知識無しに設定変更が行える。

#### 【0042】

【発明の効果】本発明によれば、多数の機器が接続された分散システムにおいて、機器の設定と機器の管理を容易にすることができます。

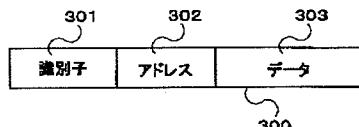
#### 【図面の簡単な説明】

【図1】本発明における第1の実施例を適応したシステムの構成図である。

【図2】第1の実施例を適応した機器の構成図である。

【図3】

図3



【図3】第1の実施例におけるメッセージのフォーマットである。

【図4】第1の実施例におけるノードの自動検出を説明する図である。

【図5】第1の実施例におけるノード管理テーブルの構成図である。

【図6】第1の実施例における各ノードに設定する構築情報である。

【図7】第1の実施例における構築ツールの処理フローである。

【図8】第1の実施例における構築ツールの処理フローである。

【図9】第1の実施例における構築ツールの処理フローである。

【図10】本発明における第2の実施例を適用したシステム構成図である。

【図11】第2の実施例におけるノード管理テーブルの構成図である。

【図12】第2の実施例における機器の設置位置を検出するための説明図である。

【図13】第2の実施例における構築ツールの処理フローである。

【図14】第2の実施例におけるユーザーが構築ツールが生成した構築情報を確認する構築情報確認画面である。

【図15】第2の実施例におけるユーザーが構築ツールが生成した構築情報を変更する構築情報変更画面である。

#### 【符号の説明】

30 100…ネットワーク、110…コントローラ、120…計算機。

【図5】

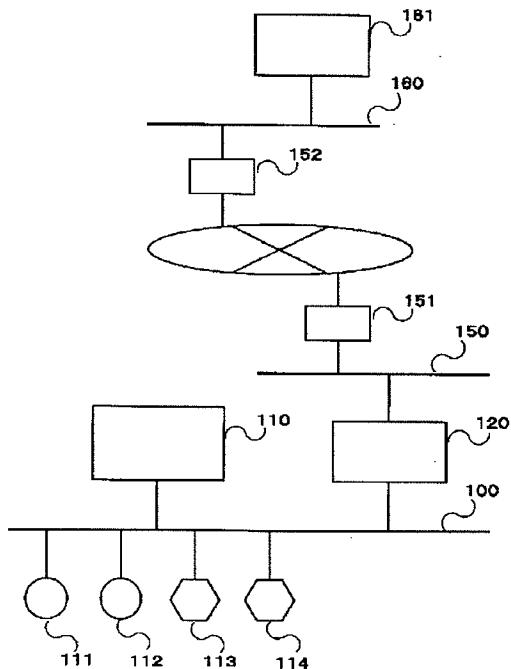
図5

| ノード<br>アドレス | 入出力<br>区分 | データ<br>長 | メー<br>カー | ファイル<br>アドレス | カタログ<br>番号 | 時間<br>間隔 | … |
|-------------|-----------|----------|----------|--------------|------------|----------|---|
| 1           | 出         | 2        | A社       | ftp://A      | A-XY3      | 200      | … |
| 5           | 出         | 2        | B社       | ftp://B      | B-EF0      | 100      | … |
| 12          | 入         | 4        | C社       | ftp://C      | C-KL2      | 300      | … |
| 33          | 入         | 4        | D社       | ftp://D      | D-WX1      | 500      | … |
| 61          | 入出        | —        | D社       | ftp://D      | D-WX1      | —        | … |
| …           |           |          |          |              |            | …        |   |
| …           |           |          |          |              |            | …        |   |
| …           |           |          |          |              |            | …        |   |

500

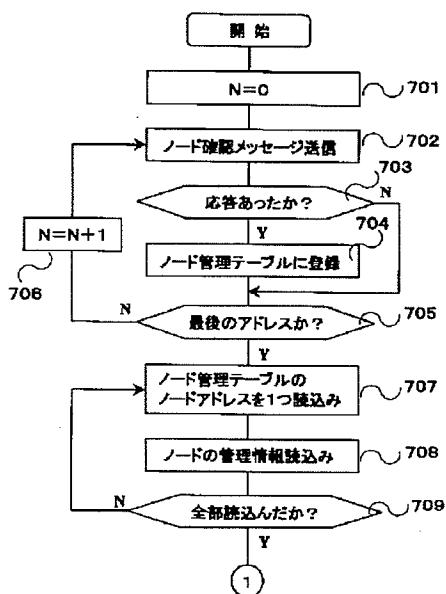
【図1】

図1



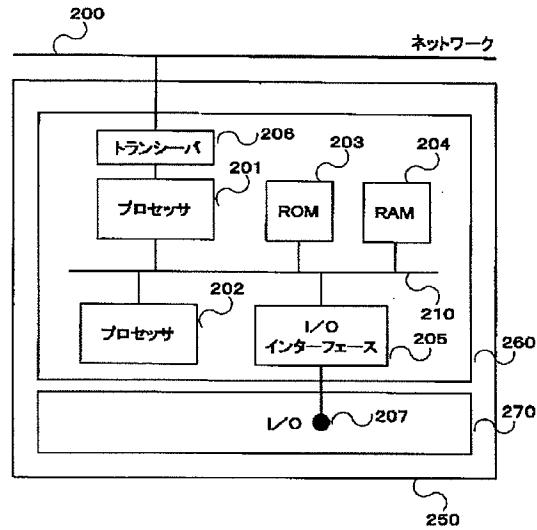
【図7】

図7



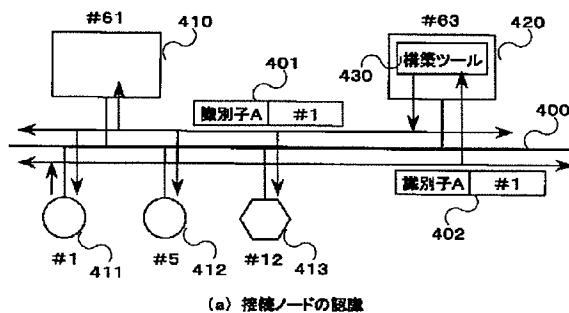
【図2】

図2

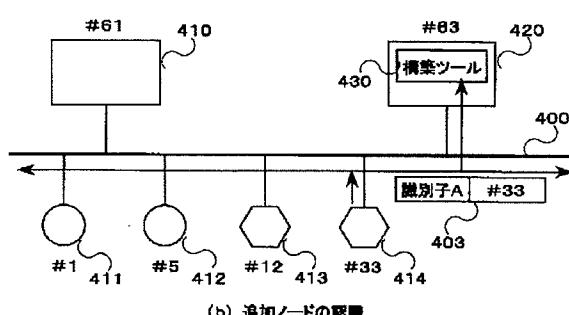


【図4】

図4



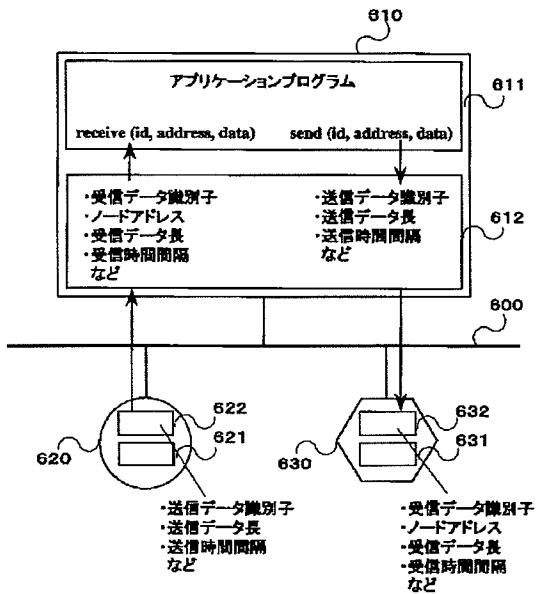
(a) 接続ノードの認識



(b) 追加ノードの認識

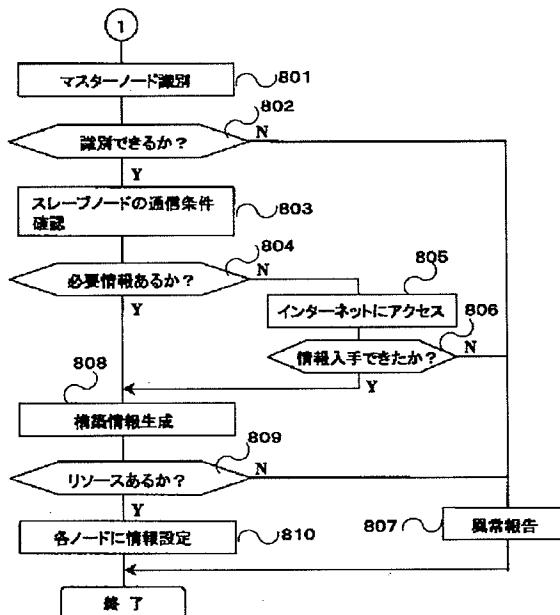
【図6】

図6



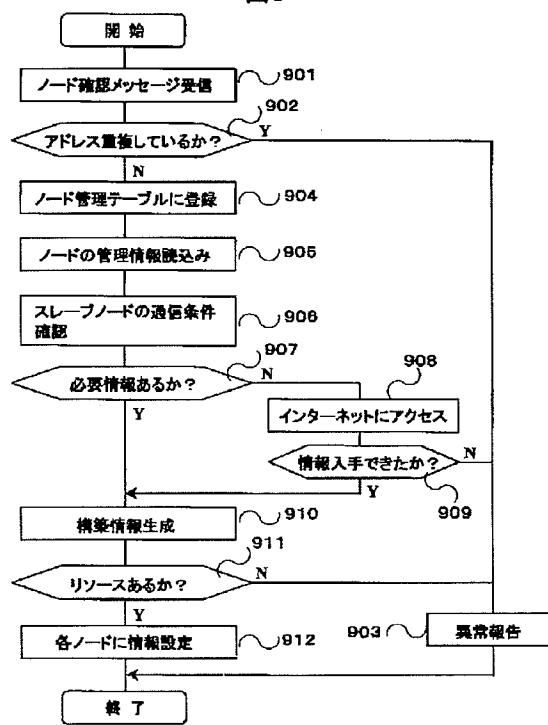
【図8】

図8



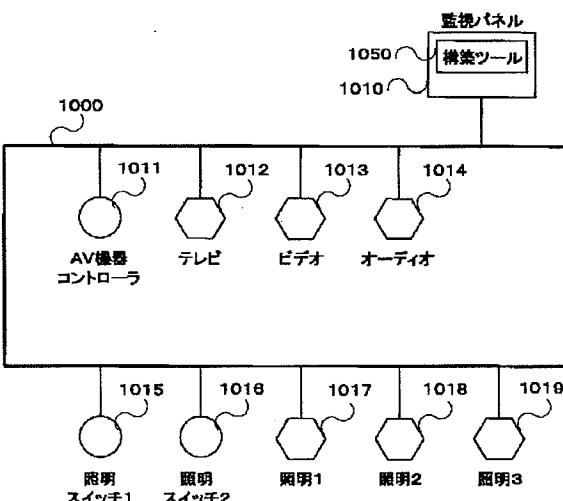
【図9】

図9



【図10】

図10



【図11】

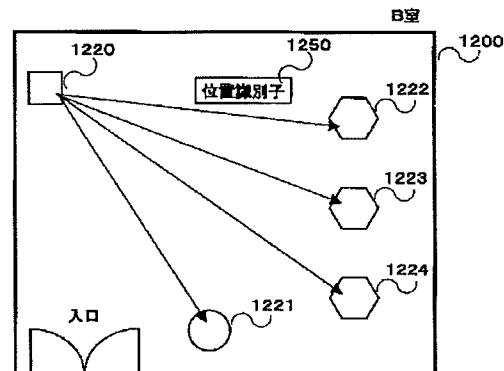
図11

| ノードアドレス | 機器種別 | 入出力区分 | データ長 | 設置位置 | メーカー | 機器名称 | ... |
|---------|------|-------|------|------|------|------|-----|
| 1059    | AV   | 出     | 4    | 1    | A社   | X    | ... |
| 2049    | AV   | 入     | 4    | 1    | B社   | Y    | ... |
| 6052    | 照明   | 入     | 1    | 3    | B社   | Z    | ... |
| 8099    | 照明   | 入     | 1    | 5    | C社   | W    | ... |
| ...     |      |       |      | ...  |      |      |     |
| ...     |      |       |      | ...  |      |      |     |

1100

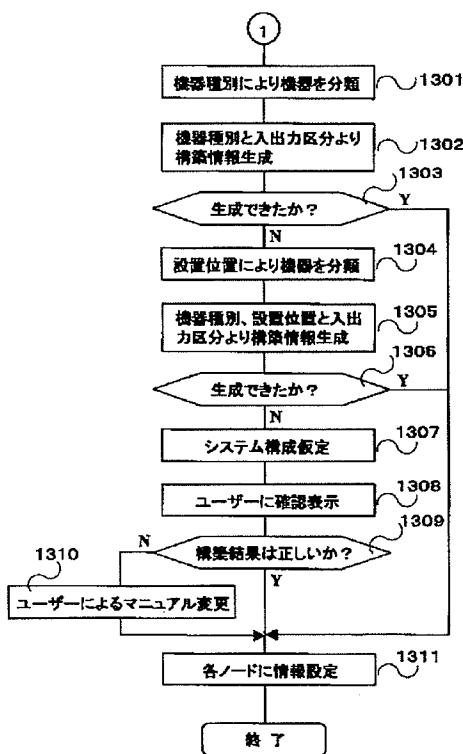
【図12】

図12



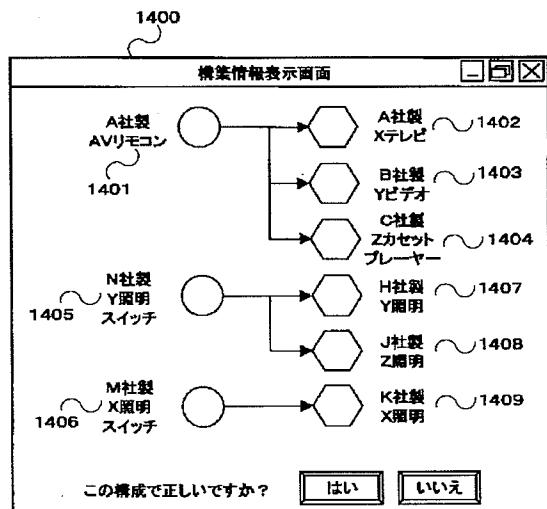
【図13】

図13



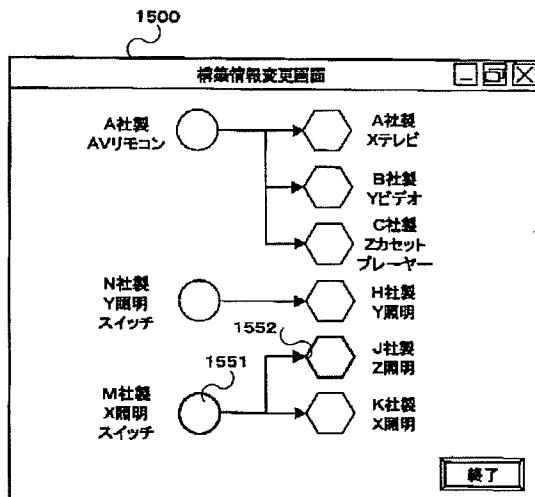
【図14】

図14



【図15】

図15




---

フロントページの続き

F ターム(参考) 5B089 GA01 GB02 JA35 JB22 KA13  
 KB04 KC44  
 5K032 BA04 EA07 EC01 EC03  
 5K033 BA04 CB13 EA07 EC01 EC03  
 5K048 BA02 BA07 DC04 DC07 EA14  
 FC01